Space Project Mission Operations Control Architecture (SuperMOCA)

SuperMOCA SYSTEM CONCEPT

Ancillary Document 3

Operations Center to Ground Terminal

Comparison of Open Protocols

July 1996

ORIENTATION

The goal of the **S**pace **Pr**oject **M**ission **O**perations **C**ontrol **A**rchitecture ("SuperMOCA") is to create a set of implementation-independent open specifications for the standardized monitor and control of space mission systems. Monitoring is the observation of the performance of the activities of these systems. Controlling is the direction of the activities performed by these systems. Overall, monitor and control is the function that orchestrates the activities of the components of each of the systems so as to make the mission work. Space mission systems include:

spacecraft and launch vehicles that are in flight, and;
their supporting ground infrastructure, including launch pad facilities and ground terminals used for tracking and data acquisition.

The SuperMOCA system concept documents consist of the following:

SuperMOCA System Concept, Volume 1: Rationale and Overview
SuperMOCA System Concept, Volume 2: Architecture
SuperMOCA System Concept, Volume 3: Operations Concepts
SuperMOCA System Concept, Annex 1: Control Interface Specification
SuperMOCA System Concept, Annex 2: Space Messaging Service (SMS) Service Specification
SuperMOCA System Concept, Annex 3: Communications Architecture
SuperMOCA System Concept, Ancillary Document 1: Ground Terminal Reference Model
SuperMOCA System Concept, Ancillary Document 2: Operations Center to Ground Terminal Scenarios
SuperMOCA System Concept, Ancillary Document 3: Operations Center to Ground Terminal – Comparison of Open Protocols

These documents are maintained by the custodian named below. Comments and questions to the custodian are welcomed.

Michael K. Jones
MS 301-235
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
Voice: 818-354-3918
FAX: 818-354-9068
E-mail: michael.k.jones@jpl.nasa.gov

Contents

SECTION **PAGE**

1. INTRODUCTION....1-1

1.1. SCOPE OF THIS DOCUMENT....1-1
1.2. CONTEXT OF TASK....1-1

2. DESCRIPTION OF CANDIDATE OPEN PROTOCOLS....2-1

2.1. COMMON OBJECT REQUEST BROKER ARCHITECTURE (CORBA)....2-1
2.1.1. Overview of CORBA....2-1
2.1.2. Organizational Source....2-5
2.1.3. Status....2-5
2.2. DISTRIBUTED COMPUTING ENVIRONMENT (DCE)....2-6
2.2.1. Overview of DCE....2-6
2.2.2. Organizational Source....2-7
2.2.3. Status....2-8
2.3. ELECTRONIC DATA INTERCHANGE (EDI)....2-8
2.3.1. Overview of EDI....2-8
2.3.2. Organizational Source....2-8
2.3.3. Status....2-8
2.4. KNOWLEDGE QUERY AND MANIPULATION LANGUAGE (KQML)....2-13
2.4.1. Overview of KQML....2-13
2.4.2. Organizational Source....2-13
2.4.3. Status....2-14
2.5. THE MANUFACTURING MESSAGE SPECIFICATION (MMS)....2-14
2.5.1. Overview of MMS....2-14
2.5.2. Organizational Source....2-17
2.5.3. Status....2-17
2.6. FIELDBUS (SP-50)....2-17
2.6.1. Overview of SP-50....2-17
2.6.2. Organizational Source....2-17
2.6.3. Status....2-17
2.7. X-WINDOWS....2-17
2.7.1. Overview of X-Windows....2-17
2.7.2. Organizational Source....2-17
2.7.3. Status....2-17

3. ASSESSMENTS OF AND RECOMMENDATIONS ON CANDIDATE OPEN PROTOCOLS....3-1

3.1. CORBA....3-1
3.2. DCE....3-2
3.3. EDI....3-2
3.4. KQML....3-2
3.5. MMS....3-3
3.6. SP-50....3-3
3.7. X-WINDOWS....3-4

4. RECOMMENDED CONFIGURATION....4-1

4.1. RECOMMENDED CONFIGURATION BY PHASE....4-1
4.2. RECOMMENDED CONFIGURATION BY INTERFACE....4-4

5. ISSUES....5-1

5.1. VIRTUAL MISSION DEVICES....5-1
6. APPENDICES....6-1
6.1. APPENDIX 1 - RECOMMENDED READING....6-1
6.2. APPENDIX 2 - DOCUMENTS LIST....6-2

Figures

FIGURE **PAGE**

FIGURE 1-1 - SUPERMOCA OPERATIONS CENTER/GROUND TERMINAL INTERFACES ..1-1
FIGURE 2-1 - CORBA ARCHITECTURE 2-3
FIGURE 2-2 - DCE ARCHITECTURE 2-6
FIGURE 4-1 - CORBA/MMS USE BY TIMELINE....4-1
FIGURE 4-2 - POSSIBLE OPERATIONS CENTER CONFIGURATION....4-2
FIGURE 4-3 - PROTOCOL USE DURING NON-CONTROL PHASES 4-3
FIGURE 4-4 - PROTOCOL USE DURING CONTROL PHASES 4-4
FIGURE 4-5 - CONTROL AND DATA FLOW MODEL WITH RECOMMENDED PROTOCOL STACKS....4-5

Tables

TABLE **PAGE**

TABLE 2-1 - X12 TRANSACTION SET INDEX VERSION 3040....2-9
TABLE 2-2 - MMS OBJECTS 2-15
TABLE 2-3 - MMS SERVICES....2-16

1. INTRODUCTION

1.1. SCOPE OF THIS DOCUMENT

This document presents a comparison of the candidate protocols based on the SuperMOCA System Concept, Ancillary Document 1, Ground Terminal Reference Model, and the SuperMOCA System Concept, Ancillary Document 2, Operations Center to Ground Terminal Scenarios. This document was developed as part of the SuperMOCA program to meet a requirement that "Candidate Open protocols, such as DCE, X Widows, EDI, KQML, the CORBA suite, MMS, SP-50, etc., will be compared on the basis of the architecture and the scenarios described in the above two documents. The comparison will focus on control and messaging protocols

1.2. CONTEXT OF TASK

A great deal of relevant work has already been accomplished within the Space Mission Operations Control Architecture (SuperMOCA) context and within the Consultative Committee for Space Data Systems (CCSDS). Figure 1-1 shows a high level view of some of this work. As shown, this Task will focus on the Management channel for the Spacecraft User, and both the Management and Transport channels for the Ground Terminal User. This document presents a comparison of the open protocols which were selected as candidates for use in this aspect of SuperMOCA. The candidate protocols are CORBA, DCE, EDI, KQML, MMS, SP-50, and X-Windows.

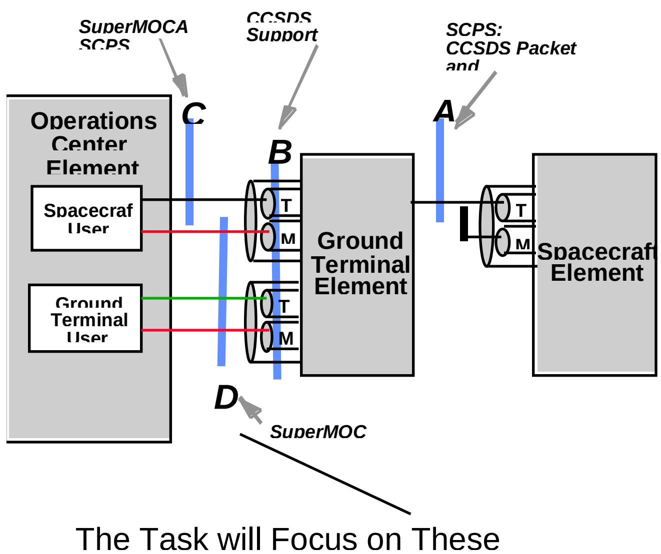

Figure 1-1 - SuperMOCA Operations Center/Ground Terminal Interfaces

2. DESCRIPTION OF CANDIDATE OPEN PROTOCOLS

2.1. COMMON OBJECT REQUEST BROKER ARCHITECTURE (CORBA)

2.1.1. OVERVIEW OF CORBA

THIS SECTION IS EXTRACTED FROM Comparing DCE and CORBA, MITRE Document MP 95B-93 (March, 1995), by Thomas J. Brando, The MITRE Corporation Distributed Object Management Integration System (DOMIS) project, funded by the Air Force Electronic Systems Center. The project office is at Rome Laboratory, Griffiss Air Force Base, Rome, New York.

CORBA supports the construction and integration of object-oriented software components in heterogeneous distributed environments. Figure 2-1 *shows the various elements that comprise the CORBA architecture.*

Object Request Broker

An Object Request Broker (ORB) provides a communication infrastructure for invoking operations on objects transparently with respect to where they are located on the network, the types of hardware and operating system platforms on which they execute, differences in data representations between platforms, the languages in which objects are implemented, and network transports used to communicate with them. CORBA specifies all of the functions that must be provided by an ORB and a set of standard interfaces to those functions. CORBA is becoming increasingly widely supported and used.

The issuance of CORBA 2.0 in December 1994 provided the specification of and mandatory requirements for inter-ORB communications functions which are the key to guaranteed interoperability. The inter-ORB protocols spelled out were the mandatory Internet Inter-ORB Protocol (IIOP), and the optional DCE Common Inter-ORB Protocol (DCE CIOP). The latter allows for implementation of a "CORBA over DCE" configuration by providing for CORBA use of the DCE Remote Procedure Call (RPC) (see below). It is the first example of the Environment-Specific Inter-ORB Protocols (ESIOPs). It is expected that there will be several ESIOPS, which will allow CORBA to operate over a variety of messaging systems. Both the IIOP and the DCE CIOP have been implemented by several vendors, have been proven to work as advertised, and are widely available.

CORBA Services

CORBA Services are services that are essential for implementing objects. The CORBA Services that have been specified thus far by OMG include:

A Concurrency Control Service that protects the integrity of an object's data when multiple requests to the object are processed concurrently.

An Event Service that supports the notification of interested parties when program-defined events occur.

An Externalization Service that supports the conversion of object state to a form that can be transmitted between systems by a means other than a request broker.

Life Cycle Services that support creation, copying, moving, and destruction of objects.

A Naming Service that permits object references to be retrieved through associations between names and objects, and for those associations to be created and destroyed.

A Persistent Object Service that supports the persistence of an object's state when the object is not active in memory and between application executions.

A Query Service that supports operations on sets and collections of objects that have a predicate-based, declarative specification and may result in sets or collections of objects.

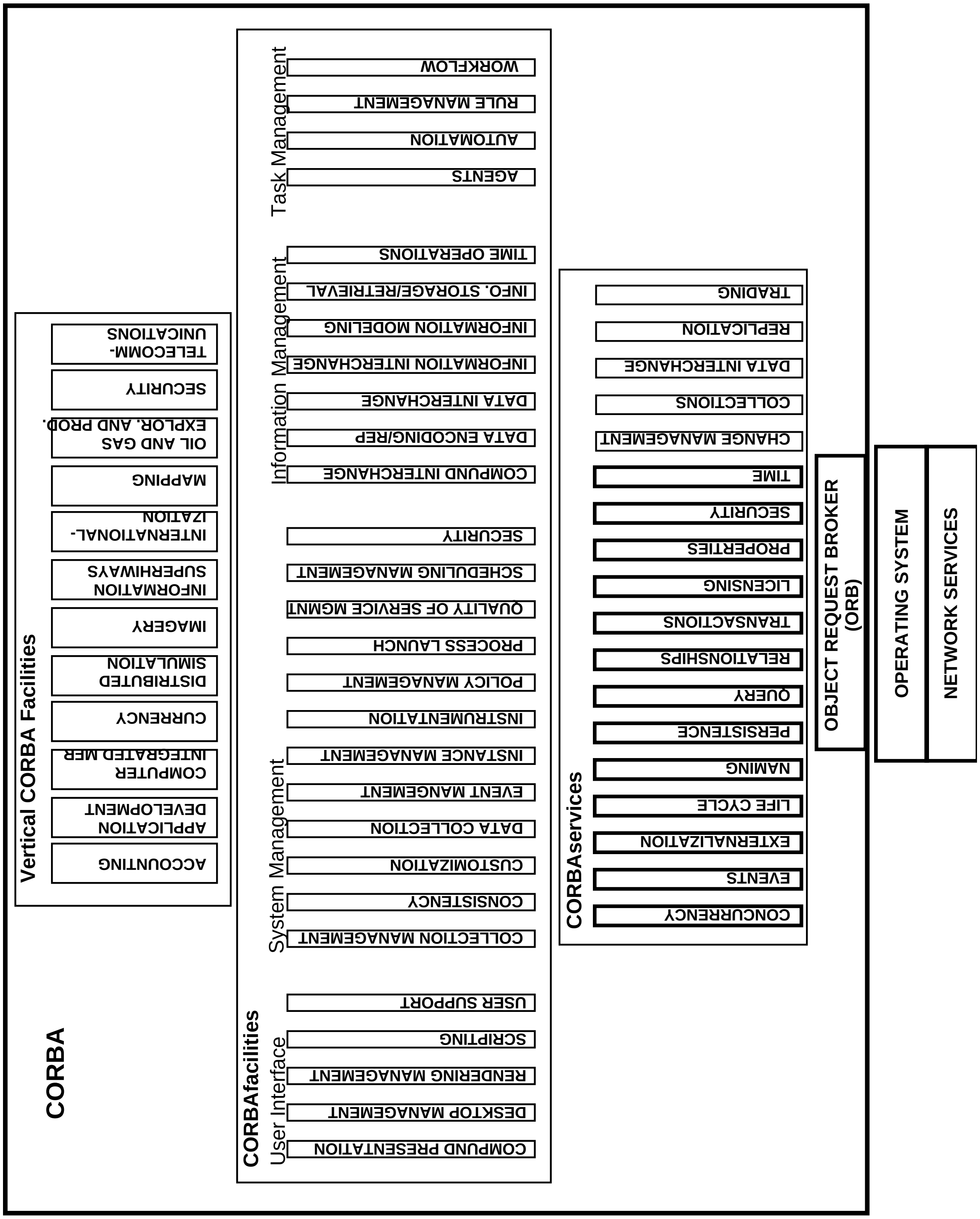

Figure 2-1 - CORBA Architecture

A Relationship Service that provides for creating, deleting, navigating, and managing relationships between objects (for example, a containment relationship between a "folder" object and the "document" objects that are considered to be "in" that folder).

A Transaction Service that provides support for ensuring that a computation consisting of one or more operations on one or more objects satisfies the requirements of atomicity (if a transaction is interrupted by a failure, any partially completed results are undone), isolation (transactions are allowed to execute concurrently, but the results are the same as if they executed serially), and durability (if a transaction completes successfully, the results of its operations are never lost, except in the event of catastrophe).

CORBA Services that are in the process of being specified and are expected to be completed in 1995 include:

A Licensing Service that will control and manage remuneration of suppliers for services rendered.

A Property Service that will support the association of arbitrary named values (the dynamic equivalent of attributes) with an object.

A Security Service that will support authentication, authorization, integrity, and privacy to degrees, and using mechanisms, that are yet to be determined.

A Time Service that will provide synchronized clocks to all objects, regardless of their locations.

CORBA Services that are not yet in the process of being specified include:

A Change Management Service that would support the identification and consistent evolution of configurations of objects.

A Collection Service that would support the creation and manipulation of collections of objects.

A Data Interchange Service that would support the exchange of data between objects.

A Replication Service that would provide for the explicit replication of objects in a distributed environment (for the purpose of availability or fault tolerance) and for the management of consistency of replicated copies.

A Trader Service that would provide a matchmaking service between clients seeking services and objects offering services.

CORBA Facilities

CORBA Facilities are useful for constructing applications across a wide range of application domains. They are divided into Horizontal CORBA Facilities, which are typically more user-oriented, and Vertical CORBA Facilities, which support specific application domains.

The Horizontal CORBA Facilities currently identified by OMG are grouped into four areas:

User Interface Facilities, which include Compound Presentation, Desktop Management, Rendering Management, Scripting, and User Support Facilities.

Information Management Facilities, which include Compound Interchange, Data Encoding and Representation, Data Interchange, Information Exchange, Information Modeling, Information Storage and Retrieval, and Time Operations Facilities.

System Management Facilities, which include Collection Management, Consistency, Customization, Data Collection, Event Management, Instance Management, Instrumentation, Policy Management, Process Launch, Quality of Service Management, Scheduling Management, and Security Facilities.

Task Management Facilities, which include Agent, Automation, Rule Management, and Workflow Facilities.

No Horizontal CORBA Facilities have been specified as yet. The first CORBA Facilities RFP was issued in October of 1994 for Compound Document Facilities (Compound Presentation and Compound Interchange). Further RFPs will be forthcoming in 1995.

The Vertical CORBA Facilities currently identified by OMG (as a result of responses from interested vertical market segments to an OMG RFI) are: Accounting, Application Development, Computer Integrated Manufacturing, Currency, Distributed Simulation, Imagery, Information Superhighways, Internationalization, Mapping, Oil and Gas Exploration and Production, Security, and Telecommunication.

No Vertical CORBA Facilities have been specified as yet.

CORBAs very rich set of proposed services (CORBAservices and CORBAfacilities) is almost an embarrassment. It seems likely that (as happened with DCE) time will see a sloughing off of some of these proposed items as user and/or vendor support for specific ones fails to materialize.

2.1.2. ORGANIZATIONAL SOURCE

The "owning" organization of CORBA is the Object Management Group (OMG), located in Framingham, Massachusetts. The OMG was formed in 1989 by eight companies as a non-profit institution, with the objective of developing guidelines, specifications, and standards to enable development and implementation of object-oriented systems in a heterogeneous environment. OMG now has over 600 members, and several international partners.

2.1.3. STATUS

CORBA is increasingly widely used and is available from most major vendors in this field. Good progress continues to be made in the specification of CORBAservices and CORBAfacilities. The major vendors in this field continue to implement and release the specified services.

2.2. DISTRIBUTED COMPUTING ENVIRONMENT (DCE)

2.2.1. OVERVIEW OF DCE

THIS SECTION IS EXTRACTED FROM Comparing DCE and CORBA, MITRE Document MP 95B-93 (March, 1995), by Thomas J. Brando, The MITRE Corporation Distributed Object Management Integration System (DOMIS) project, funded by the Air Force Electronic Systems Center. The project office is at Rome Laboratory, Griffiss Air Force Base, Rome, New York.

DCE supports the construction and integration of C-based client/server applications in heterogeneous distributed environments. Figure 2-2 *shows the various elements that comprise the DCE architecture.*

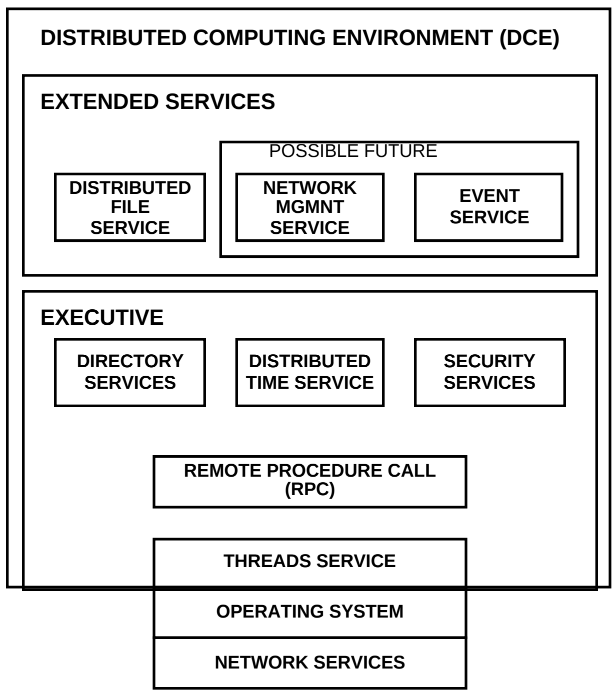

Figure 2-2 - DCE Architecture

DCE Executive

The DCE Executive consists of the following components:

Security Services that support authentication (using Kerberos V5, clients and servers can prove who they are), authorization (servers can use access control lists to determine whether a client is authorized to obtain a given service), integrity (checksums guarantee that information is received as transmitted), and privacy (DES encryption protects sensitive information from disclosure during transmission between a client and server).

Directory Services that support local DCE administration domains called cells and inter-cell name resolution. These services consist of a Cell Directory Service (CDS), *Global Directory Service (GDS, which uses X.500), Domain Name Service (DNS, not supplied, but used by DCE), and a Global Directory Agent (GDA).*

A Distributed Time Service (DTS) that synchronizes clocks on all hosts in a DCE cell, as well as between cells. DTS uses the UTC standard and is interoperable with NTP.

A Remote Procedure Call (RPC) mechanism by which clients invoke procedures in servers. A client may use directory services to bind to a particular server of interest at run time, and the client and server may use security services to guarantee desired levels of authentication, authorization, integrity, and privacy. The RPC mechanism insulates clients from details of where servers are located on the network, the types of hardware and operating system platforms on which they execute, differences in data representations between client and server platforms, and the particular network transports in use.

A Threads package based on POSIX 1003.4a (draft 4) that supports the creation and management of multiple threads of control within a client or server. A multi-threaded client may perform additional work (perhaps invoke additional RPCs) while one RPC is pending. The dispatcher that receives RPCs at a server and invokes the appropriate RPC handlers is already multi-threaded, automatically permitting DCE servers to handle multiple RPCs concurrently. The maximum number of concurrent RPCs at a server is easily configured by the developer, who is also responsible for ensuring the thread-safe behavior of all RPC handlers.

DCE Extended Services

DCE Extended Services currently consist of the Distributed File Service (DFS) alone. The DFS is a DCE application that implements a single logical filesystem that is available (through Directory Services) throughout an entire cell and across cell boundaries. DFS supports replication of files for availability and fault-tolerance and log-based recovery from hardware failures.

The remaining components that appear in Figure 2-2 *as DCE Extended Services are vestiges of OSF's Distributed Management Environment (DME), which failed to obtain vendor buy-in and has ceased to exist as an entity in its own right. These components are shown in dashed boxes because it is not clear at this time whether they will be incorporated into future releases of DCE. The Network Management Option would provide a means for management applications to access management*

information using standard network management protocols (CMIP and SNMP). The Event Service would provide a common way for system and user applications to generate, forward, filter, and log events

2.2.2. ORGANIZATIONAL SOURCE

The "owning" organization of DCE is the Open Software Foundation (OSF). The OSF is a non-profit organization headquartered in Cambridge, Massachusetts. It was formed in 1988 and has over 400 member organizations, many of them international. In addition to the facilities at Cambridge, the OSF has a large office in Grenoble, France. The purpose of the OSF is to develop, through collaborative work of its members, critical supporting open systems which are vendor-neutral, in the field of distributed systems. The OSF develops and distributes source-code for these solutions, as well as specifications and conformance tests.

2.2.3. STATUS

DCE is widely implemented and used, has a history of good performance, and is stable.

2.3. ELECTRONIC DATA INTERCHANGE (EDI)

2.3.1. OVERVIEW OF EDI

EDI is a set of internationally agreed standards, directories and guidelines for the electronic interchange of structured data, in particular that related to trade in goods and services between independent, computerized information systems.

Electronic data interchange (EDI) is for the application-to-application transfer of business documents between computers. EDI is a fast, inexpensive, and safe method of sending purchase orders, invoices, shipping notices, and other frequently used business documents. EDI is not an electronic mail message system nor is it for file sharing. Neither is it a general approach for distributed computing. With EDI it is not necessary for the communicating partners to have identical document processing systems. When a document is sent EDI translation software in the senders system translates the senders proprietary format into an EDI standard form format. The receivers EDI software automatically translates the EDI standard form into the receivers proprietary format. EDI's only purpose is to define and provide a set of standardized forms for this type of interchange.

Table 1 lists the forms currently approved by ANSI X.12. The contents of the table illustrates clearly both the scope and the purpose of EDI.

2.3.2. ORGANIZATIONAL SOURCE

The cognizant organizations for EDI are ANSI Accredited Standards Committee (ASC) X.12 in Alexandria, Virginia for the United States, and the United Nations EDIFACT group (UN/ECE in the United Nations Trade Data Interchange Directory (UNTDID)) for the international community.

2.3.3. STATUS

EDI, in both its national and international versions, is broadly and increasingly used. It is stable and implementations are widely available.

Table 2-1 - X12 Transaction Set Index Version 3040

104	Air Shipment Information
110	Air Freight Details and Invoice
125	Multilevel Railcar Load Details
126	Vehicle Application Advice
127	Vehicle Baying Order
128	Dealer Information
129	Vehicle Carrier Rate Update
130	Student Educational Record (Transcript)
131	Student Educational Record (Transcript) Acknowledgment
135	Student Loan Application
139	Student Loan Guarantee Result
140	Product Registration
141	Product Service Claim Response
142	Product Service Claim
143	Product Service Notification
144	Student Loan Transfer and Status Verification
146	Request for Student Educational Record (Transcript)
147	Response to Request for Student Educational Record (Transcript)
148	Report of Injury or Illness
151	Electronic Filing of Tax Return Data Acknowledgment
152	Statistical Government Information
154	Uniform Commercial Code Filing
161	Train Sheet
170	Revenue Receipts Statement
180	Return Merchandise Authorization and Notification
186	Laboratory Reporting
190	Student Enrollment Verification
196	Contractor Cost Data Reporting
204	Motor Carrier Shipment Information
210	Motor Carrier Freight Details and Invoice
213	Motor Carrier Shipment Status Inquiry
214	Transportation Carrier Shipment Status Message
217	Motor Carrier Loading and Route Guide
218	Motor Carrier Tariff Information
250	Purchase Order Shipment Management Document
251	Pricing Support
260	Application for Mortgage Insurance Benefits
263	Residential Mortgage Insurance Application Response
264	Mortgage Loan Default Status
270	Health Care Eligibility/Benefit Inquiry
271	Health Care Eligibility/Benefit Information
272	Property and Casualty Loss Notification
276	Health Care Claim Status Request
277	Health Care Claim Status Notification
290	Cooperative Advertising Agreements
300	Reservation (Booking Request) (Ocean)
301	Confirmation (Ocean)
303	Booking Cancellation (Ocean)
304	Shipping Instructions

309 U.S. Customs Manifest
310 Freight Receipt and Invoice (Ocean)
311 Canadian Customs Information
312 Arrival Notice (Ocean)
313 Shipment Status Inquiry (Ocean)
315 Status Details (Ocean)
317 Delivery/Pickup Order
319 Terminal Information
322 Terminal Operations Activity (Ocean)
323 Vessel Schedule and Itinerary (Ocean)
324 Vessel Stow Plan (Ocean)
325 Consolidation of Goods In Container
326 Consignment Summary List
350 U.S. Customs Release Information
352 U.S. Customs Carrier General Order Status
353 U.S. Customs Events Advisory Details
354 U.S. Customs Automated Manifest Archive Status
355 U.S. Customs Manifest Acceptance/Rejection
356 Permit To Transfer Request
361 Carrier Interchange Agreement (Ocean)
404 Rail Carrier Shipment Information
410 Rail Carrier Freight Details and Invoice
414 Rail Carhire Settlements
417 Rail Carrier Waybill Interchange
418 Rail Advance Interchange Consist
419 Advance Car Disposition
420 Car Handling Information
421 Estimated Time of Arrival and Car Scheduling
422 Shipper's Car Order
425 Rail Waybill Request
426 Rail Revenue Waybill
429 Railroad Retirement Activity
431 Railroad Station Master File
440 Shipment Weights
466 Rate Request
468 Rate Docket Journal Log
485 Ratemaking Action
490 Rate Group Definition
492 Miscellaneous Rates
494 Scale Rate Table
511 Requisition
517 Material Obligation Validation
527 Material Due-In and Receipt
536 Logistics Reassignment
561 Contract Abstract
567 Contract Completion Status
568 Contract Payment Management Report
601 Shipper's Export Declaration
602 Transportation Services Tender
622 Intermodal Ramp Activity
805 Contract Pricing Proposal
806 Project Schedule Reporting
810 Invoice
811 Consolidated Service Invoice/Statement
812 Credit/Debit Adjustment
813 Electronic Filing of Tax Return Data

815 Cryptographic Service Message
816 Organizational Relationships
818 Commission Sales Report
819 Operating Expense Statement
820 Payment Order/Remittance Advice
821 Financial Information Reporting
822 Customer Account Analysis
823 Lockbox
824 Application Advice
826 Tax Information Reporting
827 Financial Return Notice
828 Debit Authorization
829 Payment Cancellation Request
830 Planning Schedule with Release Capability
831 Application Control Totals
832 Price/Sales Catalog
833 Residential Mortgage Credit Report Order
834 Benefit Enrollment and Maintenance
835 Health Care Claim Payment/Advice
836 Contract Award
837 Health Care Claim
838 Trading Partner Profile
839 Project Cost Reporting
840 Request for Quotation
841 Specifications/Technical Information
842 Nonconformance Report
843 Response to Request for Quotation
844 Product Transfer Account Adjustment
845 Price Authorization Acknowledgment/Status
846 Inventory Inquiry/Advice
847 Material Claim
848 Material Safety Data Sheet
849 Response to Product Transfer Account Adjustment
850 Purchase Order
851 Asset Schedule
852 Product Activity Data
853 Routing and Carrier Instruction
854 Shipment Delivery Discrepancy Information
855 Purchase Order Acknowledgment
856 Ship Notice/Manifest
857 Shipment and Billing Notice
858 Shipment Information
859 Freight Invoice
860 Purchase Order Change Request - Buyer Initiated
861 Receiving Advice/Acceptance Certificate
862 Shipping Schedule
863 Report of Test Results
864 Text Message
865 Purchase Order Change Acknowledgment/Request - Seller Initiated
866 Production Sequence
867 Product Transfer and Resale Report
868 Electronic Form Structure
869 Order Status Inquiry
870 Order Status Report
872 Residential Mortgage Insurance Application
875 Grocery Products Purchase Order

876	Grocery Products Purchase Order Change
878	Product Authorization/Deauthorization
879	Price Change
880	Grocery Products Invoice
882	Direct Store Delivery Summary Information
888	Item Maintenance
889	Promotion Announcement
893	Item Information Request
894	Delivery/Return Base Record
895	Delivery/Return Acknowledgment or Adjustment
896	Product Dimension Maintenance
920	Loss or Damage Claim - General Commodities
924	Loss or Damage Claim - Motor Vehicle
925	Claim Tracer
926	Claim Status Report and Tracer Reply
928	Automotive Inspection Detail
940	Warehouse Shipping Order
943	Warehouse Stock Transfer Shipment Advice
944	Warehouse Stock Transfer Receipt Advice
945	Warehouse Shipping Advice
947	Warehouse Inventory Adjustment Advice
980	Functional Group Totals
990	Response to a Load Tender
996	File Transfer
997	Functional Acknowledgment
998	Set Cancellation

2.4. KNOWLEDGE QUERY AND MANIPULATION LANGUAGE (KQML)

2.4.1. OVERVIEW OF KQML

THE FOLLOWING IS EXTRACTED FROM KQML - A Language and Protocol for Knowledge and Information Exchange, Technical Report CS-94-02, Computer Science Department, University of Maryland, UMBC, Baltimore MD 21228, by Tim Finin, Rich Fritzson, Don McKay, and Robin McEntire

KQML is part of the ARPA Knowledge Sharing Effort which is aimed at developing techniques and methodology for building large-scale knowledge bases which are sharable and reusable. KQML is a language and a protocol that supports network programming specifically for knowledge-based systems or intelligent agents. It can be used for an application program to interact with an intelligent system or for two or more intelligent systems to share knowledge in support of cooperative problem solving.

KQML supports these protocols by making them an explicit part of the communication language. When using KQML, a software agent transmits messages composed in its own representation language, wrapped in a KQML message.

KQML is conceptually a layered language. The KQML language can be viewed as being divided into three layers: the content layer, the message layer and the communication layer. The content layer is the actual content of the message, in the programs own representation language. KQML can carry any representation language, including languages expressed as ASCII strings and those expressed using a binary notation. All of the KQML implementations ignore the content portion of the message except to the extent that they need to determine its boundaries.

The communication level encodes a set of features to the message which describe the lower level communication parameters, such as the identity of the sender and recipient, and a unique identifier associated with the communication.

The message layer forms the core of the language. It determines the kinds of interactions one can have with a KQML-speaking agent. The primary function of the message layer is to identify the protocol to be used to deliver the message and to supply a speech act or performative which the sender attaches to the content. The performative signifies that the content is an assertion, a query, a command, or any of a set of known performatives. Because the content is opaque to KQML, this layer also includes optional features which describe the content: its language, the ontology it assumes, and some type of more general description, such as a descriptor naming a topic within the ontology. These features make it possible for KQML implementations to analyze, route and properly deliver messages even though their content is inaccessible.

2.4.2. ORGANIZATIONAL SOURCE

KQML is part of the ARPA Knowledge Sharing Effort (KSE) which is aimed at developing techniques and methodology for building large-scale knowledge bases which are sharable and reusable. The KSE is a consortium to develop conventions facilitating the sharing and reuse of knowledge bases and knowledge based systems. Its goal is to define, develop, and test infrastructure and supporting technology to enable participants to build much bigger and more broadly functional systems than could be achieved working alone.

The KSE is organized around four working groups each of which is addressing a complementary problem identified in current knowledge representation technology. These groups are:

• The Interlingua Group, concerned with translation between different representation languages

• The KRSS Group (Knowledge Representation System Specification), concerned with defining common constructs within families of representation languages.

• The SRKB Group (Shared, Reusable Knowledge Bases), concerned with facilitating consensus on the contents of sharable knowledge bases

• The External Interfaces Group is concerned with run-time interactions between knowledge based systems and other modules in a run-time environment, with sub-interests in communication protocols for KB-to-KB and for KB-to-DB.

2.4.3. STATUS

The ideas which underlie the evolving design of KQML are currently being explored through experimental prototype systems which are being used to support several testbeds in such areas as concurrent engineering, intelligent design and intelligent planning and scheduling . KQML has been separately implemented by several research groups, and has been successfully used to implement a variety of information systems using different software architectures.

2.5. THE MANUFACTURING MESSAGE SPECIFICATION (MMS)

2.5.1. OVERVIEW OF MMS

MMS is a protocol at the Application Layer designed for remotely controlling and monitoring industrial devices such as programmable logic controllers, process controllers, and robots. It has a very wide range of services for device controlling and of services needed to support such control applications. It is widely used in both complex and simple applications. MMS is specified in ISO/IEC 9506-1 and -2.

The key feature of MMS is that it includes more than a communications or messaging protocol. It also defines the necessary parts (message formats, objects, functions, etc.) of a system to control and monitor the behavior of real objects.

The central concept in MMS is the Virtual Manufacturing Device (VMD). The VMD is a software object which defines the services a device can provide, a description of how a user may access and utilize those services, and a list of the variables which can be used with the device. The VMD functions as a standardizing agent between the MMS user (client) and the specific implemented device, that is, it masks the differences among specific implementations of devices so that all members of a family of similar devices appears the same to the user. The concept can be used with devices at any level from a specific hardware item, e.g. a controllable valve, to major, intelligent systems, e.g. a complex robot containing sufficient artificial intelligence that it requires only "goals" as input, rather can control instructions.

In support of the VMD concept, MMS provides a rich menu of specific services. In order to do so, MMS defines several types of objects, including those listed below in Table 2.

Table 2-2 - MMS Objects

VMD objects	The standardizing agent between the MMS user (client) and the specific implemented device
Variable and Type objects	Can be Simple or Complex
Program Control objects	Controls execution of programs within VMDs
Event objects	Reports or receives occurrences of specified actions
Semaphore objects	Controls access to resources within a VMD
Journal objects	A timebased log of data
Operator Station objects	A means of communicating with an operator
Files	Provides a file directory service and a simple means of transferring, renaming, and deleting files in a VMD

MMS operates on the above objects with a very strong set of services, shown in Table 3, below.

Table 2-3 - MMS Services

VMD Services	Initiate, Conclude, Cancel, Abort, Status, UnsolicitedStatus, GetNamList, Identify, Rename, GetCapabilityList
Variable and Type Services	Read, Write, InformationReport, GetVariableAccessAttributes, DefineNamedVariable, DefineScatteredAccess, GetScatteredAccessAttributes, DeleteVariableAccess, DefineNamedVariableList, DefineNamedType, GetNamedTypeAttributes, DeleteNamedType
Program Control Services (Domains)	InitiateDownSequence, DownloadSegment, TerminateDownSequence, InitiateUploadSequence, UploadSegment, TerminateUploadSequence, RequestDomainDownload, RequestDomainUpload, LoadDomainContent, StoreDomainContent, DeleteDomain
Program Control Services (Program Invocations)	CreateProgramInvocation, DeleteProgramInvocation, Start, Stop, Resume, Reset, Kill, GetProgrammableAttributes
Event Services	DefineEventCondition, DeleteEventCondition, GetEventConditionAttributes, ReportEventConditionStatus, AlterEventConditionMonitoring, TriggerEvent, DefineEventAction, DeleteEventAction, GetEventActionAttributes, ReportEventActionStatus, DefineEventEnrollment, DeleteEventEnrollment, GetEventEnrollmentAttributes, ReportEventEnrollmentStatus, AlterEventEnrollment, EventNotification, AcknowledgeEventEnrollment, GetAlarmSummary, GetAlaramEnrollmentSummary, AttachToEventConditionModifier
Semaphore Services	TakeControl, RelinquishControl, DefineSemaphore, DeleteSemaphore, ReportSemaphoreStatus, ReportPoolSemaphoreStatus, ReportSemaphoreEntryStatus, AttachToSemaphoreModifier
Journal Services	ReadJournal, WriteJournal, InitializeJournal, ReportJournalStatus, CreateJournal, DeleteJournal
Operator Station Services	Input, Output
Files	ObtainFile, FileOpen, FileRead, FileClose, FileRename, FileDelete, FileDirectory

The above, while obviously not providing the reader with a real understanding of MMS, does show how MMS is unique in its specific orientation towards the real-time control of devices and systems and indicates the richness of its capabilities.

2.5.2. ORGANIZATIONAL SOURCE

ISO Technical Committee 184 and the International Electrotechnical Commission (IEC) jointly are responsible for the MMS standard (ISO 9506).

2.5.3. STATUS

MMS is widely supported and used in the manufacturing field, and that use is increasing. There are many vendors of MMS compatible devices and systems. MMS is mature and stable.

2.6. FIELDBUS (SP-50)

2.6.1. OVERVIEW OF SP-50

TEXT TO BE SUPPLIED

2.6.2. ORGANIZATIONAL SOURCE

2.6.3. STATUS

2.7. X-WINDOWS

2.7.1. OVERVIEW OF X-WINDOWS

TEXT TO BE SUPPLIED

2.7.2. ORGANIZATIONAL SOURCE

2.7.3. STATUS

3. ASSESSMENTS OF AND RECOMMENDATIONS ON CANDIDATE OPEN PROTOCOLS

3.1. CORBA

CORBA is becoming increasingly widely supported and used. The issuance of CORBA 2.0 in December 1994 provided the specification of and mandatory requirements for inter-ORB communications functions which are the key to guaranteed interoperability. The inter-ORB protocols spelled out were the mandatory Internet Inter-ORB Protocol (IIOP), and the optional DCE Common Inter-ORB Protocol (DCE CIOP). The latter allows for implementation of a "CORBA over DCE" configuration by providing for CORBA use of the DCE Remote Procedure Call (RPC) (see below). It is the first example of the Environment-Specific Inter-ORB Protocols (ESIOPs). It is expected that there will be several ESIOPS, which will allow CORBA to operate over a variety of messaging systems. Both the IIOP and the DCE CIOP have been implemented by several vendors, have been proven to work as advertised, and are widely available.

CORBAs very rich set of services (CORBAservices and CORBAfacilities) offers distinct advantages for SuperMOCA applications, especially since there will undoubtedly be a significant learning curve for SuperMOCA. The availability of the broad range of services would offer flexibility and a "reserve" capability which may well be needed. In fact, the richness of CORBA services and facilities is almost an embarrassment. It seems likely that (as happened with DCE) time will see a sloughing off of some of these proposed items as user and/or vendor support for specific ones fails to materialize.

Another strength of CORBA for SuperMOCA is that it is specifically designed to support Object Oriented applications. It seems clear that SuperMOCA will heavily use Object Oriented techniques, and it is also evident that there will be a need to utilize "legacy" software systems. Object Oriented approaches are very effective in interfacing to legacy software.

The greatest weaknesses of CORBA are that its use is not yet as widespread as some of the other possibilities, e.g. DCE, and that it is still developing, that is many of its services and facilities are not yet either specified or implemented. However, the currently specified and available set of services is, as noted above in the Overview of CORBA, already broad and very useful. In addition, several vendors have implemented CORBA over DCE implementations which allow the use to take advantage of both CORBA and DCE simultaneously, and this is enhance by the availability of the DCE CIOP. Further, the other CORBA services and facilities are being actively worked and use of CORBA is growing rapidly. In addition, there is some evidence of increased interest and activity in an eventual convergence of CORBA and DCE.

The broad present and future services and facilities of CORBA, its Object Orientation, and the availability of CORBA over DCE implementations lead to the recommendation that it be used as the primary open protocol for the non-control phases of the SuperMOCA Operations Center/Ground Terminal operations (Planning, Scheduling, part of Initiation, part of Termination, and Evaluation). At least initially implementations should probably include CORBA over DCE, migrating as deemed advisable to CORBA only, or if it comes to pass, to an eventual merged CORBA/DCE.

3.2. DCE

DCE is widely accepted and used for all types of distributed computing implementations. Its services are well established, stable, and reliable and are available from a broad base of companies. Further, the services which are provided are essential for this SuperMOCA application. They are an inter-communication method - the Remote Procedure Call (RPC) - and the Security, Directory, Time, and Distributed File Services. As with CORBA, DCE is Transport independent, and as with CORBA, DCE uses an Interface Definition Language.

The weaknesses of DCE are that its service array is very limited, activity to increase the number of services appears to be very slow, and, although some adaptations are being made, it was not designed to support Object Oriented systems. Perhaps the most significant current activity related to DCE is the "CORBA over DCE" implementations of several vendors as mentioned in the assessment of CORBA, above.

The strengths of DCE - including reliability and availability - are sufficiently great that its adoption by SuperMOCA for the Operations Center/Ground Terminal interface is recommended. Its strengths can be utilized and its weaknesses can be overcome by using a "CORBA over DCE" implementation as noted above. Also as noted above, this solution is especially appealing as there is some initial motion towards an eventual merging of the two.

3.3. EDI

EDI is in wide and increasing usage both nationally and internationally. It is extensively used in the manufacturing, wholesaling, distribution, retailing, financial, medical and many other industries. However, it is fundamentally a forms based communication method for these industries without the characteristics of distributed computing and systems controlling which are basic needs in the SuperMOCA context. The applicability of EDI to the Operations Center/Ground Terminal needs is very limited. While it could be used in some aspects of the Planning and Scheduling processes it seems unlikely that advantages gained would be worth the incorporation of the protocol in all the necessary sites. Use of EDI by SuperMOCA in this area is not recommended.

3.4. KQML

KQML is a protocol/environment targeted at increasing the understanding of the requirements for inter-communication among distributed intelligent systems. Its object is to eventually lead to standardization of inter-communication among such cooperating systems. As such, it is basically a research tool, and is not a "real" candidate for SuperMOCA at this time. However, the progress of KQML should be tracked by SuperMOCA. As Ground Terminals and their systems become more and more intelligent KQML or a descendant of KQML may become desirable and even essential. The timelines of KQML development and of "intelligent" Ground Terminals and Operations Centers may be compatible.

3.5. MMS

MMS is in wide and increasing use in the equipment control field. An implementation experience base exists in industry and is rapidly broadening. Furthermore, the abilities and characteristics of MMS appear to be an excellent match to the SuperMOCA needs in the Ground Terminal control area, between the controlling systems and physical devices, or between controlling systems and VMDs, whether the VMDs are at the equipment, subsystem or system level. There does not appear to be any real competitor to MMS in this field. MMS is specifically designed for these kinds of applications, and therefore does not have strong capabilities in the more general areas of distributed computing such as file access and sharing and the kinds of interactions needed in Planning, Scheduling and parts of Initiation, parts of Termination, and in the Maintenance phase. While MMS could be used in these areas, that would require a significant amount of development of specialized agreement on how to so use it.

Of particular interest to SuperMOCA is the use which has already been made of MMS in the control and monitoring of JPLs DSS-13. This effort was very successful and beneficial. Reports on this effort are available as SuperMOCA Papers.

It is recommended that MMS be used by SuperMOCA as the open protocol between the Operations Center controlling systems and the physical devices of the Ground Terminal, and/or between the Operations Center controlling systems and the VMDs of the Ground Terminal, at whatever level the VMDs are available. It seems clear that MMS should also be used between different levels of VMDs within the Ground Terminal (e.g. between a system level VMD and the subsystem level VMDs of which it is constituted). However, the latter recommendation may be outside the scope of this task. It is not recommended that MMS be used as a general open protocol in the non-control portions of Operations Center/Ground Terminal processes. More specifically, it is recommended that MMS be the active protocol during parts of the Initiation and Termination phases and all of the Operations phase.

3.6. SP-50

SP-50 shares many of the characteristics of MMS. It is basically a subset of MMS with special characteristics to increase speed of control and monitoring and to enhance use in hazardous environments. It is focused on use in a LAN environment, and its use is and probably will continue to be less widespread than MMS. It is not clear that the use of or adaptation of SP-50 in the Operations Center/Ground Terminal interface offers any significant advantage over the use of MMS, and in fact the less widespread use of SP-50 and its LAN orientation are disadvantages. However, SP-50 is obviously a strong candidate for use onboard spacecraft (outside the scope of this document), and may also be a strong candidate for use within Ground Terminals between different levels of VMDs and/or between VMDs and their physical devices (also outside the scope of this document). It is not recommended that SP-50 be used by SuperMOCA in the Operations Center/Ground Terminal interface.

3.7. X-WINDOWS

X-Windows is basically a remote terminal system, not a true distributed computing environment. Although it can be made to work for distributed computing, it would be an inefficient and awkward approach since that was not its design goal. Relative to the SuperMOCA Operations Center/Ground Terminal interface problem, X-Windows is a non-starter and its use is not recommended in this context.

4. RECOMMENDED CONFIGURATION

4.1. RECOMMENDED CONFIGURATION BY PHASE

The recommended configuration is to use CORBA (initially over DCE) for the non-control phases of Operations Center/Ground Terminal activities, and to use MMS for the control phases. Whether CORBA should be used for the non-control parts of the Initiation and Termination sub-phases is for further study. Figure 4-1 illustrates the two possibilities.

Figure 4-2 represents a possible software configuration for an Operations Center, using the recommended protocols. The figure shows the two basic kinds of stacks, one for general distributed computing use, and the other for real-time operation and monitoring. Note that the only difference between the stacks in the lower layers is the (possible) use of UDP instead of TCP for the Control/Monitor protocols.

Figure 4-3 illustrates the protocol usage in terms of the Operations Center and the Ground Terminal Reference Model during the non-controlling phases, i.e., Planning, Scheduling, the non-controlling parts of Initiation and Termination, and Evaluation.

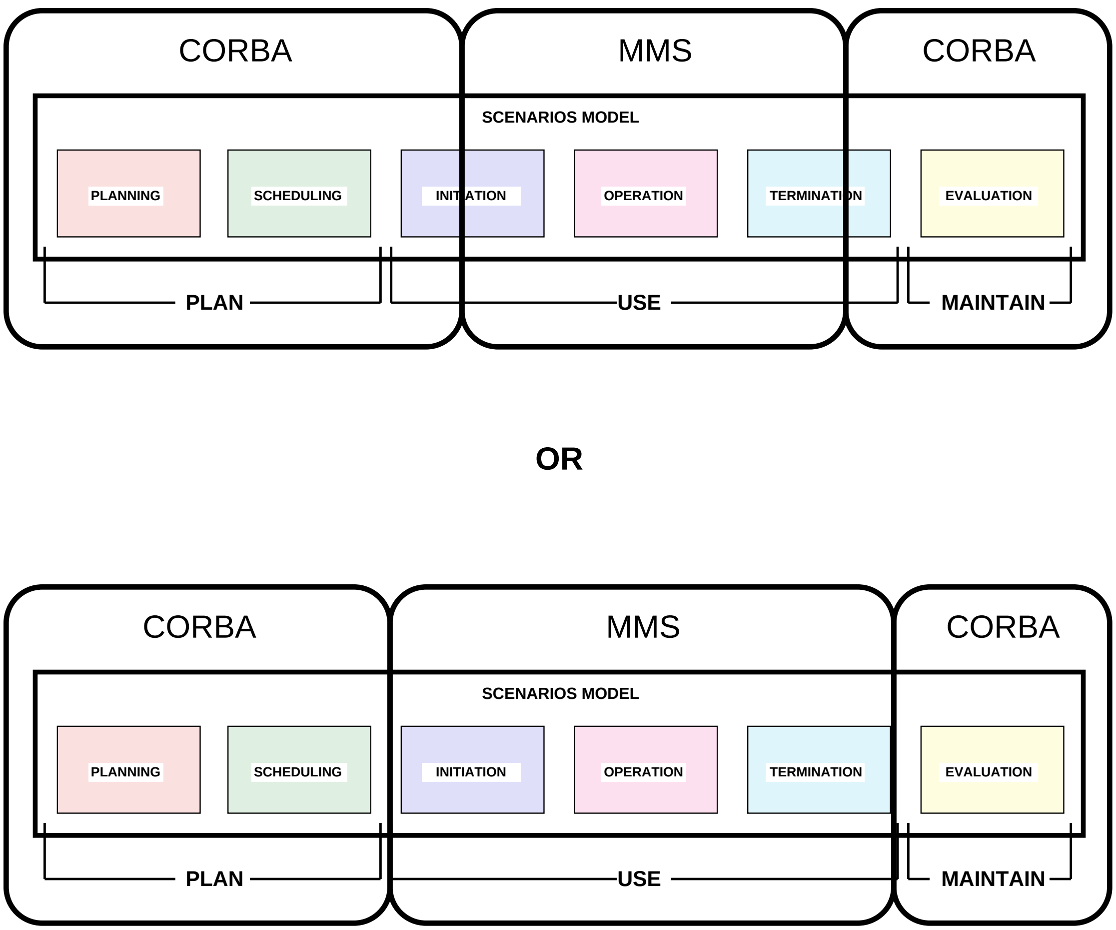

Figure 4-1 - CORBA/MMS Use by Timeline

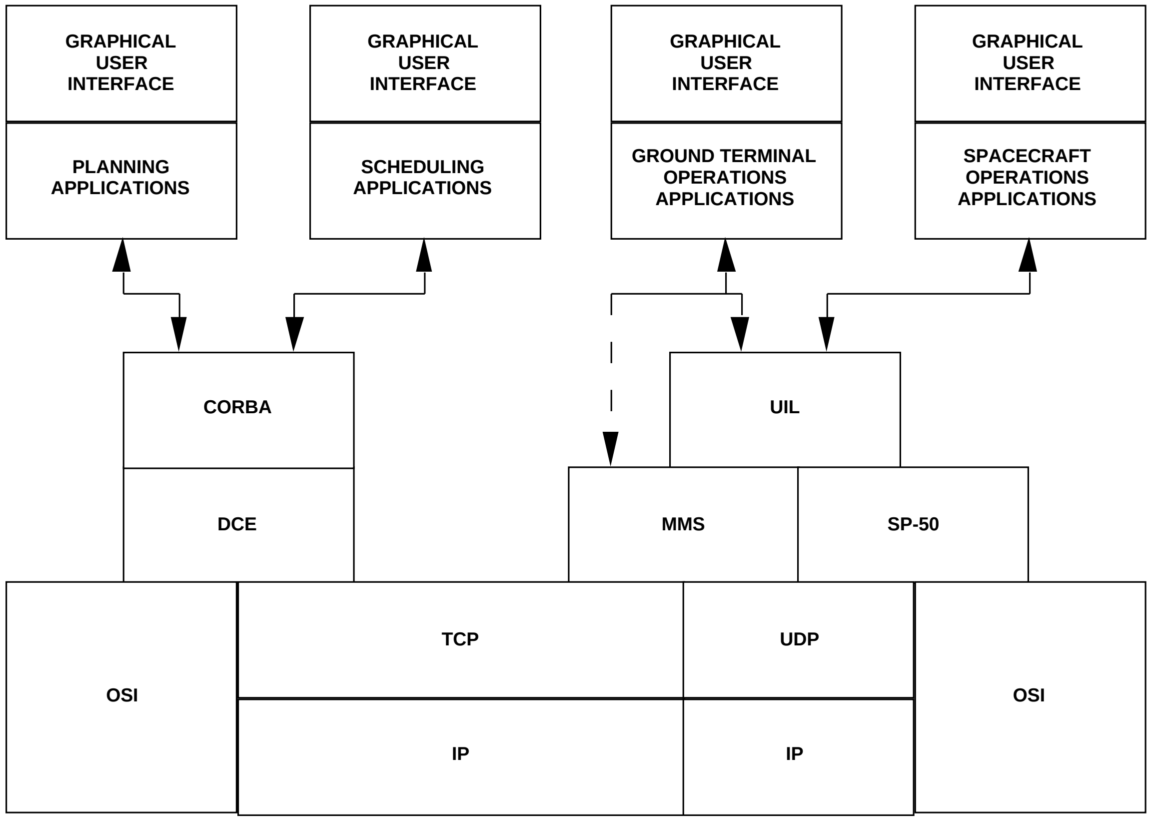

Figure 4-2 - Possible Operations Center Configuration

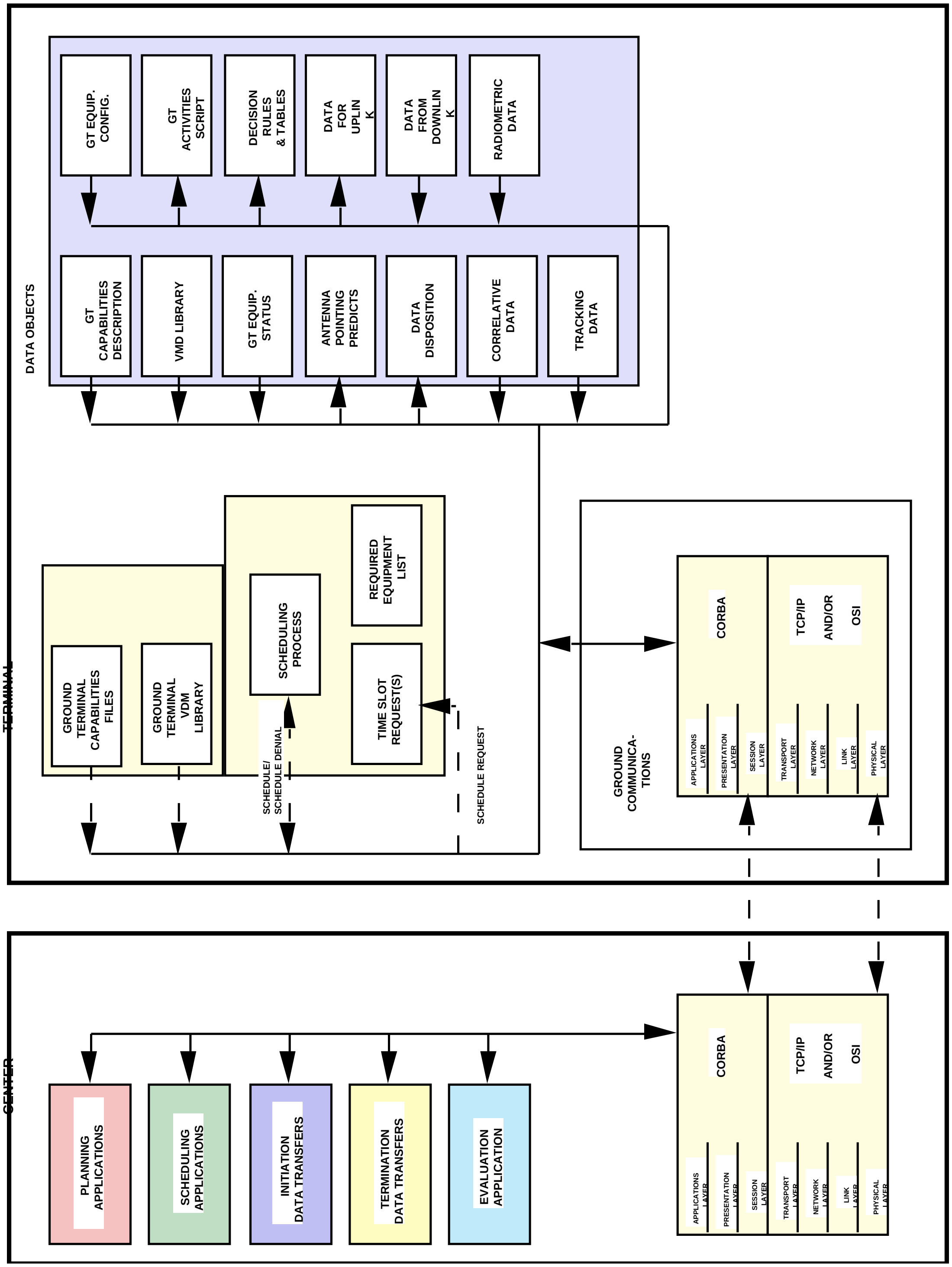

Figure 4-3 - Protocol Use During Non-Control Phases

Adding the Operations Center, Spacecraft, and recommended protocols to the Ground Terminal Reference Model System Level drawing yields Figure 4-4 below showing the protocols in use during the process of controlling the Spacecraft and Ground Terminal. Peer to peer communications is also shown.

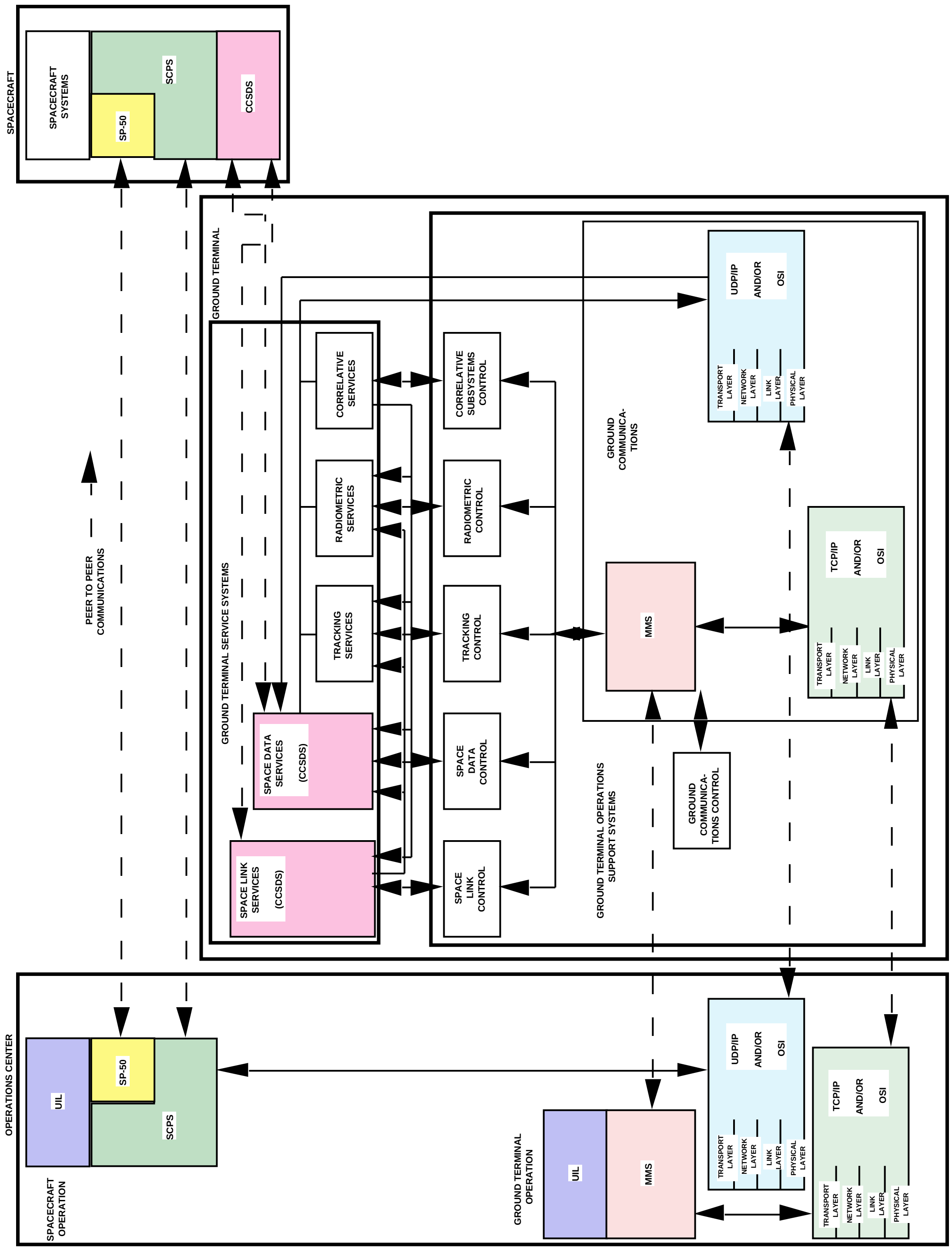

Figure 4-4 - Protocol Use During Control Phases

4.2. RECOMMENDED CONFIGURATION BY INTERFACE

Adding the recommended protocol stacks to the Ground Terminal Reference Model Control and Data Flow Model produces Figure 4-5, below.

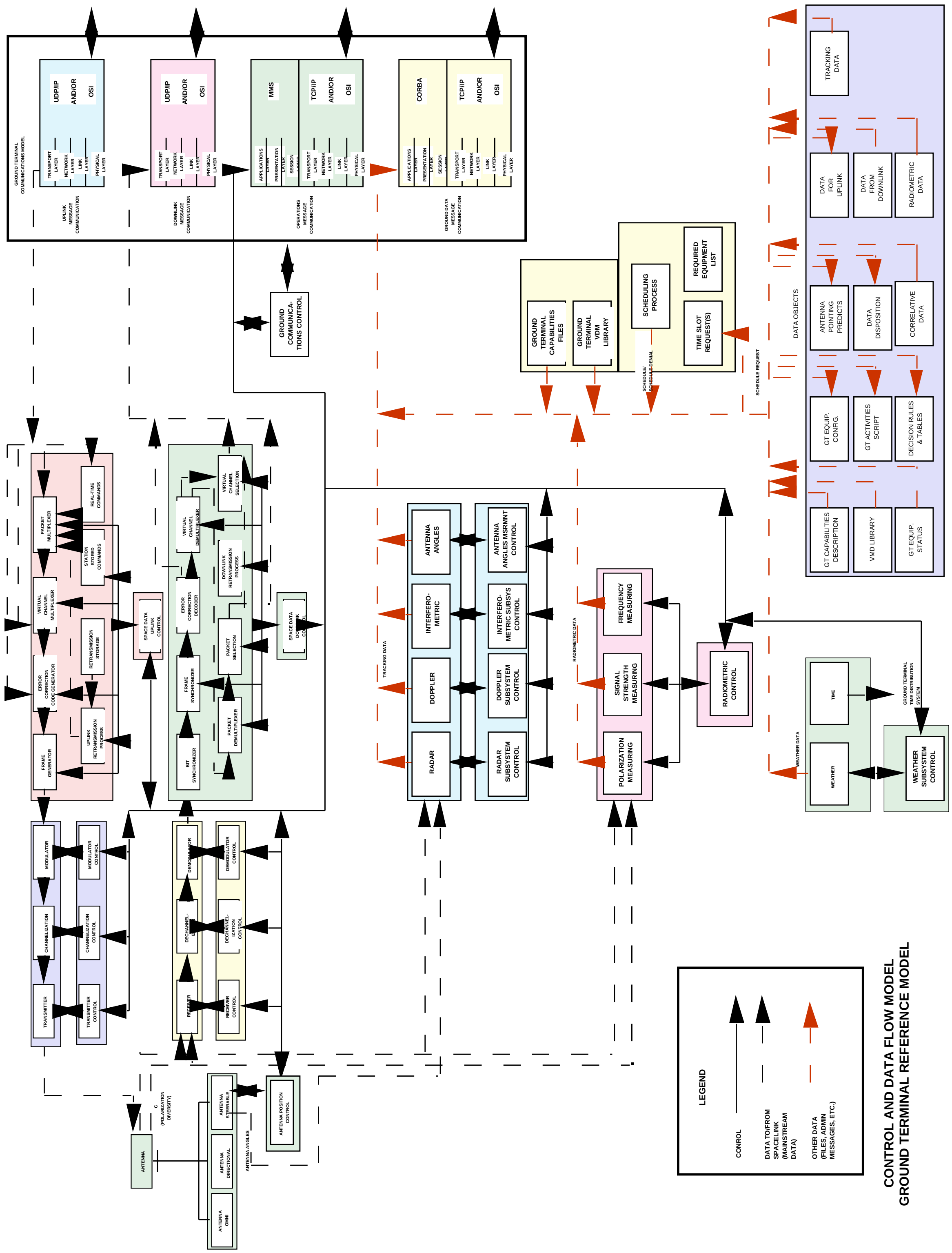

Figure 4-5 - Control and Data Flow Model with Recommended Protocol Stacks

5. ISSUES

5.1. VIRTUAL MISSION DEVICES

VMDs appear to be a, if not the, central issue for interoperability implementation strategies. The development of VMDs would be the enabling technology for any kind of standardized cross-agency Operations Center/Ground Terminal interactions. While it is conceivable that an Operations Center could utilize/operate a previously unknown Ground Terminal by accessing the low-level equipment control instruction repertoire and linking it to the Operations Center's systems, it would appear to be a very difficult methodology. On the other hand, if a set of VMD interfaces were developed in which (a subset of?) the control input side was standardized, masking the low-level equipment control instruction repertoire, then the problem appears to be of reasonable difficulty.

Furthermore, the develop of a common VMD set would greatly enhance the development of intelligent Ground Terminal subsystems, systems, and terminals. The development of a broadly usable common set of Rules would be greatly simplified if the reasoning engine had only to deal with a known, general, family of equipment with common behavior.

The actual location of the VMD software is not important (at least at this level). It could be at the Operations Center, at the Ground Terminal interface, or even within the Ground Terminal, with intelligent subsystem and system interfaces to the Operations Center.

What is important, is that the open protocols selected for Operations Center/Ground Terminal interoperability be capable of supporting any of these distributed operations.

6. APPENDICES

6.1. APPENDIX 1 - RECOMMENDED READING

The author of this document found the two following documents extremely useful and informative. They are highly recommended to anyone interested in the CORBA, DCE, and/or MMS protocols:

Comparing DCE and CORBA, MITRE Document MP 95B-93 (March, 1995), by Thomas J. Brando, The MITRE Corporation Distributed Object Management Integration System (DOMIS) project, funded by the Air Force Electronic Systems Center. The project office is at Rome Laboratory, Griffiss Air Force Base, Rome, New York. Available on the World Wide Web at:

http://www.mitre.org/research/domis/reports/DCEvCORBA.html

and

Overview and Introduction to the Manufacturing Message Specification, Document DWG# 100286/1.0, Systems Integration Specialists Company, Inc. (SISCO), Sterling Heights, MI. Available on the World Wide Web at:

http://www.sisconet.com/techinfo.htm

6.2. APPENDIX 2 - DOCUMENTS LIST

The following is a partial list of the documents reviewed in preparation of this report.

CORBA DCE	Comparing DCE and CORBA	Thomas J. Brando	MITRE	3/95
CORBA	Interoperability and the CORBA Specification	Thomas J. Brando	MITRE	2/95
CORBA	CORBA Components	DOMIS Project	MITRE	95
CORBA	CORBA Services	DOMIS Project	MITRE	95
CORBA	HP Introduces CORBA 2.0-Compliant HP Distributed Small Talk	Press Release	Hewlett-Packard Company	8/95
CORBA	Coupling Finite Element Codes Using CORBA-Based Environments	Deborah Kernan	Sandia National Laborotories	1/96
CORBA	Benefits of Using Objects Request Brokers (ORB's)	Expersoft	Expersoft Corporation	96
CORBA	Inter-Language Unification	Xerox PARC	Xerox Corporation	5/96
DCE	A Distributed Computing Environment Framework: An OSF Perspective	Brad Curtis Johnson	Open Software Foundation	6/10/91
DCE	DCE and Objects	David Chappell	Chappell and Associates	3/96
DCE	The OSF Distributed Computing Environment: Building on International Standards	O-DCE-WP19-1	Open Software Foundation	4/92
DCE	Distributed Computing Environment	OSF-DCE-PD-1090-4	Open Software Foundation	1/92
DCE	NASA's Mission to Planet Earth: Changing the World's Environment	OSF-NASA-CS-1294	Open Software Foundation	95
DCE	"Tuxedo"/DCE; Providing Tuxedo-to-DCE Integration		Novell, Inc.	95

EDI	The Business and Information Modelling Framework for UN/EDIFACT	Colin Clark	T5 BIM Group	9/95
EDI	X.12 Transaction Set Index Version 3040			
EDI	New Information Technologies and Their Impact on How We Do Business	Andre Valerand	President, EDI World Institute	95
EDI	DoD Electronic Data Interchange			
EDI	EAN and EDI	Au Soo Keun	Federation of Malaysian Manufacturers	95
KQML	An Architecture for Information Agents	Donald P. McKay, J. Pastor and R. McEntire	Loral Defense Systems	96
KQML	KQML - A Language and Protocol for Knowledge and Information Exchange	Tom Finan and Rich Fritzon, UMBC, and D. McKay and R. McEntire, Unisys	Computer Science Dept., University of Maryland Baltimore Campus, and Unisys	
KQML	KQML Applications			
KQML	Knowledge Sharing Effort			
KQML	Evaluation of KQML as an Agent Communication Language	James Mayfield, Y. Labrou, T. Finan	Computer Science Dept., University of Maryland Baltimore Campus	96
KQML	KQML Public Directory	Tom Finan	Computer Science Dept., University of Maryland Baltimore Campus	96
X-Windows	Multiplatform X-Windows	Charles W. Veth, President	CVM, Inc.	11/93

X-Windows				
MMS	Overview and Introduction to the Manufacturing Message Specification (MMS)		Systems Integration Specialists Company, Inc. (SISCO)	6/94
MMS	An Overview to the Manufacturing Message Specification	Ralph Mackiewicz		94
MMS	Support of MMS Services by Major Vendors	?	?	6/29/94
MMS	An OSI Based Architecture for Tracking Station Automation	W. Randy JHeuser, Mike Stockett, Richard Chen	JPL	
MMS	Industrial Protocols for Spacecraft Command and Control	W. Randy Heuser	JPL	
Multi-Protocol	Open Solutions to Distributed Control in Ground Tracking Stations	W. Randy Heuser	JPL	
UIL	A Next Generation Language for Space Mission Operations	Randy L. Davis	University of Colorado	95
General	Recommendations for Standards Required to Achieve Space Data Network Interoperability	Adrian J. Hooke	JPL	
Cross-support	CCSDS Cross Support System Description, Volume 1, Concept and Scenarios		CCSDS	90